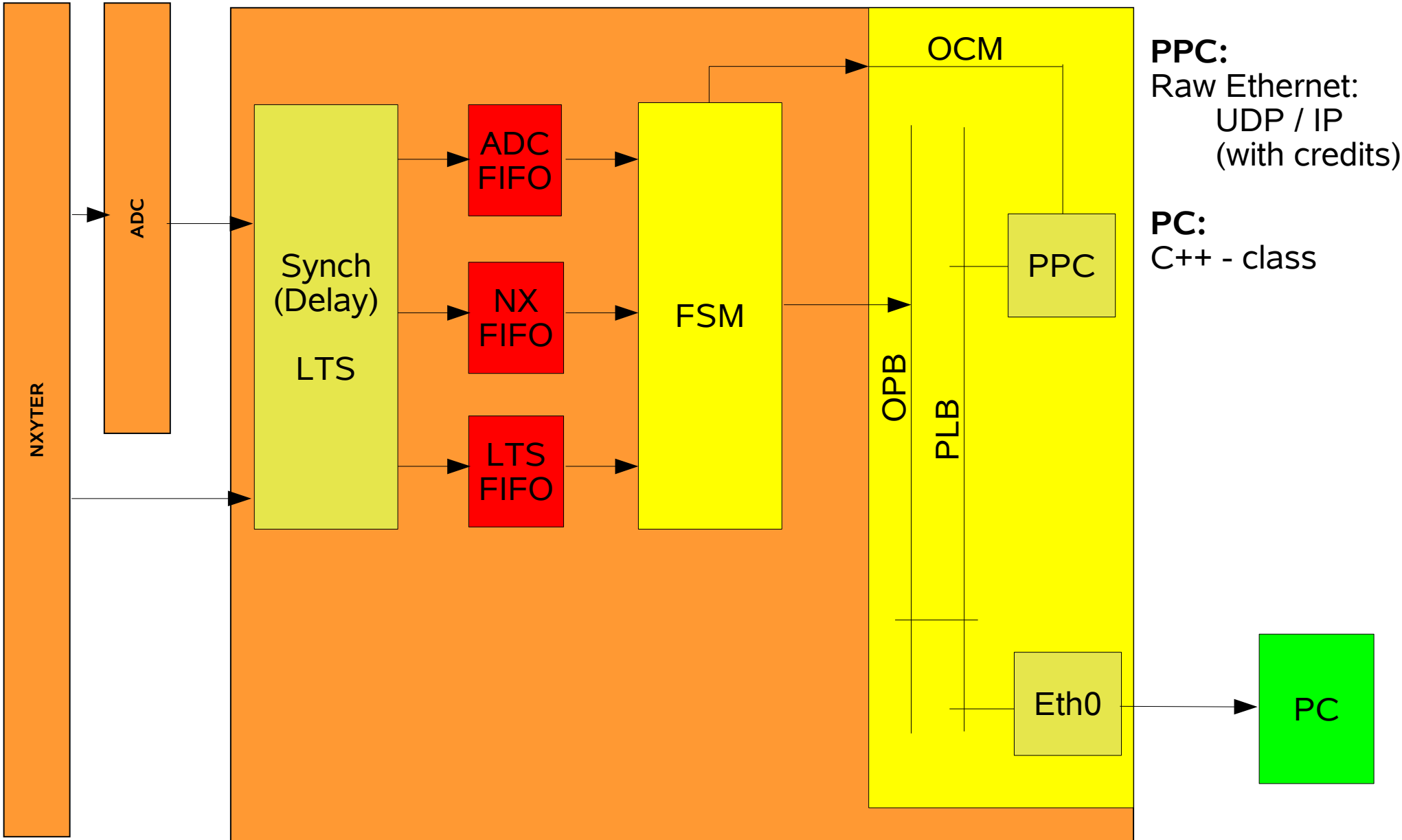
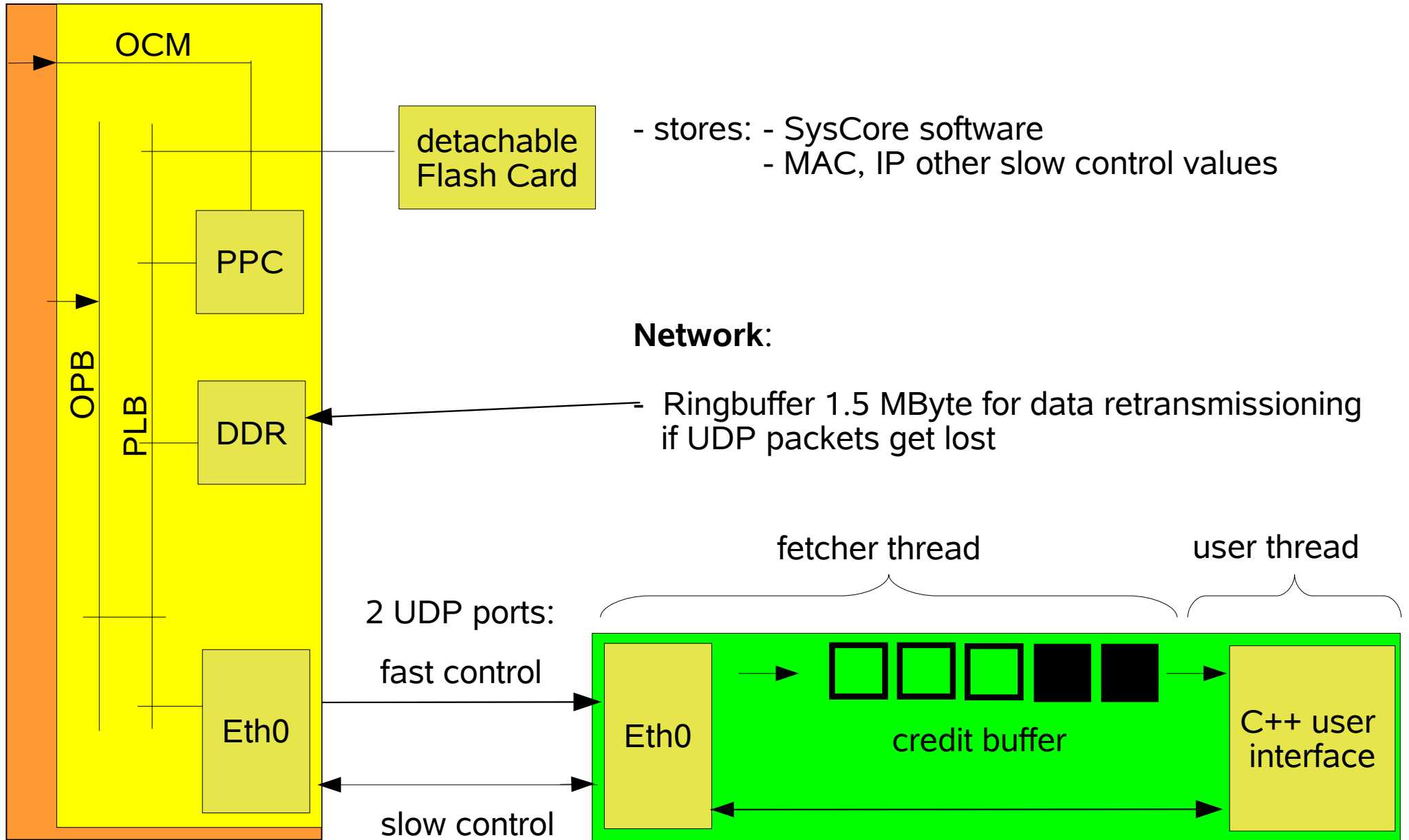


KIP Heidelberg

Stefan Müller-Klieser

Status of the ROC interface class





SysCoreControl class header

```
class SysCoreControl
{
private:
    ...

public:
    vector<SysCoreBoard> board;

    SysCoreControl();
    ~SysCoreControl();

    /**
     * Slowcontrol
     */
    void addBoard(ipaddr_t address);
    void addBoard(string address);
    int getNumberOfBoards();

    void peek(unsigned short boardId, unsigned long address, unsigned long &readValue);
    void poke(unsigned short boardId, unsigned long address, unsigned long writeValue);

    //Debug "telnet" connection
    void sendConsoleCommand(unsigned short boardId, string cmd);

    /**
     * Fastcontrol
     */
    bool isDataInBuffer();
    void getNextData(SysCoreData &sysCoreData);
};
```

```
...  
  
#include "SysCoreControl.h"  
  
int main(void)  
{  
    SysCoreControl control;  
  
    control.addBoard("129.206.177.44");  
    control.addBoard("129.206.177.45");  
    ...  
  
    cout << control.getNumberOfBoards();  
    cout << " SysCore boards are configured." << endl;  
  
    //Slowcontrol  
  
    control.board[0].setIpAddress("129.206.177.100");  
  
    //or  
    ipaddr_t ip;  
    Utility::u2ip("129.206.177.101", ip);  
    control.poke(1, SC_SET_IPADDRESS, ip);  
  
    control.board[0].startDAQ();  
    cout << "starting readout..." << endl;  
  
    ...  
}
```

```
...
SysCoreData data;

//read 100 hitdatas

for(int i = 0; i < 100;)
{
    if(control.board[0].isDataInBuffer()) //credit buffer has data
    {
        control.board[0].getNextData(data);

        //handle data

        switch (data.getMessageType())
        {
            case ROC_MSGTYPE_NOP:
                break;
            case ROC_MSGTYPE_HITDATA:
                i++;
                break;
            ...
        }
    }
}
```

- Class SysCoreData has a private char[6] array
- .getter() use bitmasks and bitshifting

- finish implementing all SysCoreData.getter() for every Messagetype naming defines for the address space, etc.
(1/2 day)
- Merging the Network design and the Flash Card storage design
(1 day)
- Merge with the bootloader design
(1 day)